Chapter 1

Getting Acquainted with Linux

In This Chapter

- Diving into Linux origins and features
- Telling Linux apart from the rest of the operating system pack
- Depending on GNU and the GPL
- Marveling at the Linux Company (or lack thereof)
- Checking out popular Linux distributions

Ford, you're turning into a penguin. Stop it!

— Arthur Dent

Welcome to the world of Linux, the operating system developed by more than a thousand people around the world! In this chapter, you find out about Linux itself — what it is, where it comes from, and why it gets so much attention in the news these days. Prepare to have your assumptions challenged about how software *must* be developed and sold, and open your mind to new possibilities.

COPYRIGHTED MATERIAL

Is Free Really Free?

Understanding Linux requires a radical shift of thought regarding the way that you acquire and use computer software. (***Note:*** By *radical,* we mean getting to the root of the matter, rather than putting on beads and camping out in the administration building.) Your first step toward shifting your mind-set is to alter your general connotation of the word *free* to represent *freedom,* rather than *free lunch.* That's right; you can sell "free" software for a fee . . . and you're encouraged to do so, as long as you relay the same freedom to each recipient of the software.

Don't scratch your head too hard; these concepts are tough to grasp initially, especially when you consider the conditioning you've received from the commercial software industry's marketing departments. Perhaps you don't know that when you purchase most proprietary, shrink-wrapped software, you don't actually *own* the software. Rather, you're granted permission to use the software within the bounds dictated by the licensor.

Linux also has a license. However, the motives and purpose of the license are much different from those of most commercial software. Instead of using a license to restrict use of the software, the GNU General Public License (GPL) that Linux uses ensures that the software will always be open to anyone. No company can ever own or dictate the way in which you use or modify Linux — although they can have their own individual copyrights and trademarks on their various brands of it, such as Red Hat and Novell. In essence, you already own Linux, and you can use it for anything you like, as long as you propagate the GPL freedoms to any further recipients of the software.

Linux: Revolution or Just Another Operating System?

Before going any farther into Linux, we need to get some terminology out of the way.

An *operating system* is the software that runs your computer, handling all interactions between you and the hardware. Whether you're writing a letter, calculating a budget, or managing your recipes on your computer, the operating system provides the essential air that your computer breathes. Furthermore, an operating system isn't just one program; it consists of hundreds of smaller programs and utilities that allow us humans to use a computer to do something useful. You then run other programs (such as your word processor) on top of the operating system to get everything done.

Linux has been accused of being "just another operating system." On the surface, it may appear so, but if you look deeper, you can see that this isn't so. The Linux project is a flagship leading the current trend toward open-source and free (as in "freedom," not "free beer") software within the computing industry. A rock-solid operating system because of the model under which it was (and continues to be) developed, Linux represents much that is good in software development.

Two fundamental distinctions separate Linux from the rest of the operating-system pack:

- Linux is licensed under the unique and ingenious *GNU General Public License,* which you can read about in the next section.
- Linux is developed and maintained by a worldwide team of volunteer and paid programmers working together over the Internet.

Linux is great for many reasons, including the fact that the folks who built it from the ground up wanted it to be all the following:

- **Multiuser:** More than one user can be logged in to a single computer at one time.
- **Multiprocesser:** True *pre-emptive multitasking* enables the operating system's core to efficiently juggle several programs running at once. This is important for providing multiple services on one computer.
- **Multiplatform:** Linux currently runs on more than 24 *platforms* (hardware types), including 32- and 64-bit Intel-based PCs, Digital/Compaq Alpha, all variants of the Apple Macintosh, Sun SPARC, the Apple iPod, and even the Microsoft Xbox.
- **Interoperable:** Linux plays nice with most network protocols (languages) and operating systems, allowing you to interact with users and computers running Microsoft Windows, UNIX, Novell NetWare, Macintosh computers, and other groups that occupy smaller market niches.
- **Scalable:** As your computing needs grow, you can rely on Linux to grow with you. The same Linux operating system can run on a tiny electronic photo frame, a desktop computer, or a very large, industrial-strength server system.
- **Portable:** Linux is mostly written in the C programming language. *C* is a language created specifically for writing operating-system-level software and can be readily *ported* (translated) to run on new computer hardware.
- **Flexible:** You can configure the Linux operating system as a network host, router, graphical workstation, office productivity PC, home entertainment computer, file server, Web server, cluster, or just about any other computing appliance you can think of.

- **Stable:** The Linux *kernel* (the heart of the operating system) has achieved a level of maturity that makes most software developers envious. It's not uncommon to hear reports of Linux servers running for years without crashing.
- **Efficient:** The modular design of Linux enables you to include only the components needed to run your desired services. Even older Pentium computers can utilize Linux and become useful again.
- ***Free!:*** To most people, the most intriguing aspect of Linux is the fact that it's often available free of charge. How (the capitalists murmur) can anyone build a better mousetrap with no incentive of direct monetary return?

In this chapter, we intend to answer that last question for you. We also hope to paint a picture of the open-source software development model that created Linux.

So where did Linux come from?

Although programming of the Linux core started in 1991, the design concepts were based on the time-tested *UNIX* operating system. UNIX was developed at Bell Telephone Laboratories in the late 1960s. The original architects of UNIX, working back when there were few operating systems, wanted to create an operating system that shared data, programs, and resources both efficiently and securely — an ideal that wasn't available then (and is still sought after now). From there, UNIX evolved into many different versions; its current family tree is so complicated that it looks like a kudzu infestation.

In 1991, Linus Torvalds was a computer science student at the University of Helsinki in Finland. He wanted an operating system that was like the UNIX system that he'd grown fond of at the university, but both UNIX and the hardware it ran on were prohibitively expensive. A UNIX version called Minix was available for free, but it didn't quite meet his needs. So, Torvalds studied Minix and then set out to write a new version himself. In his own words (recorded for posterity on the Internet because this was in an early version of an online chat room), his work was "just a hobby, won't be big and professional like GNU."

Writing an operating system is no small task. Even after six months of hard work, Torvalds had made very little progress toward the general utility of the system. He posted what he had to the Internet — and found that many people shared his interest and curiosity. Before long, some of the brightest minds around the world were contributing to Linus's project by adding enhancements or fixing bugs (errors in the code).

Anatomy of an Open-Source Software Project

To the casual observer (and some corporate IT decision makers), Linux appears to be a freak mutation. How, after all, can something so complex and discipline-dependent as a computer operating system be developed by a loosely knit band of volunteer computer geeks from around the world?

Just as science is constantly attempting to classify and explain everything in existence, technology commentators are still trying to understand how the open-source approach can create superior software, especially in cases where no one is in charge. Often the reasons have much to do with the usual human desire to fill a need with a solution. When a programmer in the Linux world wants a tool, the programmer simply writes one — or bands together with other people who want a similar package, and they write it together.

GNU who?

Imagine software created out of *need* rather than projected profit. Even though UNIX ultimately became proprietary software, the motives for its creation were originally based on practical needs. What people usually refer to as *the Linux operating system* is actually a collection of software tools created with the express purpose of solving specific computing problems.

The speed of Linux's popularity also wouldn't be possible without the vision of Richard Stallman. Massachusetts Institute of Technology (MIT) has long held a reputation for nurturing the greatest minds in the technological disciplines. In 1984, Stallman, a gifted student and brilliant programmer at MIT, was faced with a dilemma — sell his talent to a company for a tidy sum of money or donate his gifts to the world.

Stallman set out on a journey to create a completely free operating system that he would donate to the world. He understands — and continues to live — the original hacker ethic, which declares that information wants to be free. This concept wasn't new in his time. In the early days of the computing industry, many advancements came from freely sharing ideas and programming code. Manufacturer-sponsored user groups brought the best minds together to solve complicated problems. This ethic, Stallman felt, was lost when companies began to hoard software as their own intellectual property with the single purpose of profit.

As you may or may not have gathered by this point, widespread and accessible source code is paramount to successful software development. *Source code* is the term for the human-readable text (as opposed to the unreadable cyber-hieroglyphics in an "executable" file) that a programmer types to communicate instructions to the computer.

Writing computer programs in binary is an extremely arduous task. Modern computer software is usually written in a human-friendly language and then *compiled,* or translated, into the computer's native instruction set. To make changes to this software, a programmer needs access to a program's source code. Most proprietary software comes only as a precompiled product; the software developer keeps the source code for those programs under lock and key.

After determining that his operating system would be built around the conceptual framework of UNIX, Stallman wanted the project name to distinguish his system from UNIX. So he chose the recursive acronym *GNU* (pronounced ga-*new*), which means "GNU's not UNIX."

To finance the GNU project, Stallman organized the Free Software Foundation (FSF), which sold free (that is, open-source) software to help feed the programmers who worked on its continuing development. (Remember, we're talking *free* as in *free speech,* not as in *free beer.*) Although this organization (and its goal of creating a complete operating system) was necessary and important, a much more important piece of the puzzle had to be put into place. The new software needed a license to protect it from big-business pirates — a concern still all too relevant today as a former Linux company tries to hijack ownership of decades of volunteer work from thousands of people around the world.

The *GNU General Public License* (GPL) is a unique and creative software license that uses copyright law to protect the freedom of the software user, which is usually the opposite of how a copyright works. Generally, a copyright is an enforceable designation of ownership and restriction from duplication by anyone but the copyright holder. When software is licensed under the GPL, recipients are bound by copyright law to respect the freedom of anyone else to use the software in any way they choose. Software licensed with the GPL is also known as copy*left* software (the reverse of *right*, get it?). Another way to remember the GPL is through its ultimate result: Guaranteed Public for Life.

While Stallman's work set the stage for Linux's rapid climb to popularity, the operating system he and his crew were working on took longer than expected. If you're interested in the completed version, go to `www.gnu.org/software/hurd/hurd.html`.

Who's in charge of Linux anyway?

As an open-source project evolves, various people emerge as leaders. This leader is known as the project's *benevolent dictator.* The benevolent dictator has probably spent more time than anyone else on a particular problem and often has some unique insight. Normally, the words *democratic* and *dictator* are never paired in the same sentence, but the open-source model is a very democratic process that endorses the reign of a benevolent dictator.

Linus Torvalds is still considered the benevolent dictator of the Linux kernel (the operating system's core). He ultimately determines what features are added to the kernel and what features aren't. The community trusts his vision and discretion. In the event that he loses interest in the project, or the community decides that he has gone senile, a new leader will emerge from amongst the very competent people working with him.

Einstein was a volunteer

Someone who is a volunteer or donates time to a project isn't necessarily providing a second-rate effort (or only working on weekends and holidays). In fact, any human-resources expert will tell you that people who choose to do a job of their own free will produce the highest-quality products.

The volunteers who contribute to open-source projects are often leaders in their fields who depend on community collaboration to get useful work done. The open-source concept is no stranger to the scientific community. The impartial peer-review process that open-source projects foster is critical in validating some new feature or capability as being technically correct.

Those who paint the open-source community as copyright violators and thieves often misunderstand — or outright ignore — these vital issues. Open-source programmers are very proud of their work *and* are also very concerned about their own copyrights, not wanting their work to be stolen by others — hence licenses such as the GPL. This concern creates an atmosphere with the greatest respect for copyright. Bandits who claim that they're "just being open-source" when they steal other people's hard work are grossly misusing the term to soothe their own consciences.

Many have also pointed out that if copyright is violated in open source, it's easy to tell. Watch the news and notice how often large software corporations are convicted of stealing other people's code and incorporating it into their own work. If the final product is open-source, it's easy for anyone to look and make sure nothing stolen is in it. As you might imagine, tracking down such copyright violations is much more difficult in a closed-source scheme.

Packaging Linux: The Distribution

A complete Linux system package is called a *distribution*. A Linux distribution contains the Linux kernel, the GNU project's tools, and any number of open-source software projects to provide diverse functionality for the system. By combining all the pieces into one package, you don't have to go hunting for the individual pieces across the Internet.

There are lots of different Linux distributions available to meet just about any computing requirement you could have. Most distributions are customized for specific user groups — such as business users, multimedia enthusiasts, software developers, or normal home users.

Each customized distribution includes the software packages required to support specialized functions, such as audio- and video-editing software for multimedia enthusiasts, or compilers and Integrated Development Environments (IDEs) for software developers.

The different Linux distributions are often divided into three categories:

- Core Linux distributions (more about the core in a minute)
- LiveCD test distributions
- Specialized distributions

The following sections describe these different types of Linux distributions, and show some examples of Linux distributions in each category.

A single Linux distribution often appears in several different versions to cover more ground. For example, Fedora releases both a full core distribution, as well as a LiveCD version that contains a subset of the full system. Many of the specialized Linux distributions (such as Ubuntu) are based on the Debian core Linux distribution. Ubuntu uses the same installation files as Debian, but packages only a small fraction of a full-blown Debian system.

Core Linux distributions

A *core* Linux distribution contains the Linux and GNU operating systems, one or more graphical desktop environments, and just about every Linux application that is available, ready to install and run. The core Linux distribution provides one-stop shopping for a complete Linux installation, no matter what your requirements are. Table 1-1 shows some of the more popular core Linux distributions out there.

Table 1-1 Core Linux Distributions

Distribution	*Where to Find It*	*Description*
Slackware	`www.slackware.com`	One of the original Linux distribution sets, popular with Linux geeks.
Red Hat	`www.redhat.com`	A commercial business distribution used mainly for Internet servers.
Fedora	`www.fedoraproject.org`	A spinoff from Red Hat, designed for home use.
Gentoo	`www.gentoo.org`	A distribution designed for advanced Linux users, containing only Linux source code.
Mandriva	`www.mandriva.com`	Designed mainly for home use (previously called Mandrake).
openSuSe	`www.opensuse.org`	Different distributions for business and home use (now owned by Novell).
Debian	`www.debian.org`	Popular with Linux experts and commercial Linux product developers.

We use the Fedora Linux distribution in this book to illustrate how to work with a core Linux distribution.

In the early days of Linux, a distribution was released as a set of floppy disks. You had to download groups of files and then manually copy them onto disks. It would usually take 20 or more disks to make an entire distribution! Needless to say, this was a painful experience. Nowadays, with home computers commonly having CD and DVD players built in, Linux distributions are released as either a CD set or a single DVD. This makes installing Linux much easier.

The Linux LiveCD

While having lots of options available in a distribution is great for Linux geeks, it can become a nightmare for beginning Linux users. Most distributions ask a series of questions during the installation process to determine which applications to load by default, what hardware is connected to the computer, and how to configure the hardware. Beginners can often find these questions confusing. As a result, they often either load way too many programs on their computers

or don't load enough and later discover that their computers won't do what they want them to do. Fortunately for beginners, there's a much simpler way to install Linux.

A relatively new phenomenon in the Linux world is the bootable Linux CD distribution, called a *LiveCD.* Most modern computers can start up by reading the operating system from the CD instead of the hard drive. This arrangement lets you see what a Linux system is like without actually installing it.

To take advantage of this feature, some Linux distributions create a bootable CD that contains a sample Linux system. Due to the limitations on the single CD size, the sample can't contain a complete Linux system, but you'd be surprised at all the software they can cram on there! The result is pretty cool: You can boot your computer from the CD and run a Linux distribution without having to install anything on your hard drive!

This is an excellent way to test various Linux distributions without having to mess with your PC. Just pop in a CD and boot! All the Linux software will run directly off the CD. There are lots of Linux LiveCDs that you can download from the Internet and burn onto a CD for a test drive. Table 1-2 shows some popular Linux LiveCDs that are available.

Table 1-2 Linux LiveCD Distributions

Distribution	*Where to Find It*	*Description*
Knoppix	`www.knoppix.net`	A German Linux distribution, the first Linux LiveCD developed.
SimplyMEPIS	`www.mepis.org`	Designed for beginning home Linux users.
PCLinuxOS	`www.pclinuxos.com`	Full-blown Linux desktop workstation on a single CD.
Ubuntu	`www.ubuntu.com`	A worldwide Linux project, designed for many languages.
Slax	`www.slax.org`	A live Linux CD based on a subset of the Slackware Linux distribution.
Puppy Linux	`www.puppylinux.org`	A full-featured Linux distribution designed for older computers.

Some Linux LiveCD distributions, such as Ubuntu, allow you to install the Linux distribution directly from the LiveCD. This enables you to boot with the CD, test drive the Linux distribution, and then if you like it, install it onto your hard drive. This feature is extremely handy and user friendly.

We use the Ubuntu LiveCD distribution in this book to demonstrate using LiveCDs.

As with all good things, Linux LiveCDs have a few drawbacks. Because you access everything from the CD, applications run more slowly, especially if you're using older, slower computers and CD drives. Also, because you can't write to the CD, any changes you make to the Linux system will be gone the next time you reboot.

But advances are being made in the Linux LiveCD world that help to solve some of these problems:

- Copying Linux system files from the CD to memory to run faster
- Copying system files to a file on the hard drive to save them
- Storing system settings on a USB flash drive
- Storing user settings on a USB flash drive

Some Linux LiveCDs, such as Puppy Linux, are designed with a minimum number of Linux system files; they're copied directly into memory when the CD boots — and you can remove the CD from the computer as soon as Linux is done booting. Not only does this make your applications run much faster (because applications run faster in computer memory), it frees up your CD tray so you can use it for ripping audio CDs or playing video DVDs from the software included.

Specialized Linux distributions

Over the past few years, a new subgroup of Linux distributions has started to appear. These are typically based on one of the core distributions, but contain only a subset of applications that would make sense for a specific area of use.

Besides providing specialized software (for example, bundles of office products for business users), customized Linux distributions also attempt to help beginning Linux users by auto-detecting and auto-configuring common hardware devices. This makes installing Linux a much more enjoyable process.

Table 1-3 shows some of the specialized Linux distributions available and what they specialize in.

Table 1-3 Specialized Linux Distributions

Distribution	*Where to Find It*	*Description*
Linspire	`www.linspire.com`	A commercial Linux package configured to look like Windows.
Xandros	`www.xandros.com`	A commercial Linux package preconfigured for beginners.
SimplyMEPIS	`www.mepis.org`	A desktop distribution for home users.
Ubuntu	`www.ubuntu.com`	A desktop and server distribution for school and home use.
PCLinuxOS	`www.pclinuxos.com`	A distribution focusing on home and office use.
gOS	`www.thinkgos.com`	A distribution designed for beginners, and pre-installed on some laptop and desktop systems.
Puppy Linux	`www.puppylinux.org`	A small distribution that runs well on older PCs.

That's just a small sampling of specialized Linux distributions. There are hundreds of specialized Linux distributions, with more popping up all the time on the Internet. No matter what your specialty, you'll probably find a Linux distribution made for you.

We use the gOS distribution in this book to see how some of the features in a specialized Linux distribution work. The gOS distribution includes the Google Desktop applet, which we discuss in Chapter 15.